An Obstacle Detection System Using Depth Information and Region Growing for Blind

[*1] Ching-Tang Hsieh, [1] Wei-Ming Lai, [1] Cheng-Hsiang Yeh, [12] Hsieh-Chang Huang
[1,] *Department of Electrical Engineering, Tamkang University*
New Taipei, Taiwan, R.O.C.
[2,] *Department of Information Technology, Lee-Ming Institute of Technology*
New Taipei, Taiwan, R.O.C.
[*] *Correspondending Author: hsieh@ee.tku.edu.tw*

Abstract

In order to make the visually impaired people get the information of obstacles effectively and avoid it successfully in the unfamiliar environment, we propose an obstacle detection method based on depth information. Firstly, we use the edge characteristics of depth image to segment the obstacle by different depth. Then we remove the unnecessary ground information by gradient threshold. Our algorithm can label the obstacles by region growing algorithm respectively. Finally, we use rectangular windows to box out these obstacles. Our algorithm can display distances between Kinect sensor and these centers of obstacles on the frame for accuracy. Experimental results show that the proposed method has greater robustness than others. And the average of processing speed is only 0.08 second per frame.

Keywords*: Obstacle detection; Kinect; Depth image; Travel aid*

1. Introduction

There are so many visually impaired people relying on the guide cane or guide dogs to move around freely in the world. But, not every visually impaired people can easily pair successfully and obtain guide dogs, they often have to wait for a long time. In addition, visually impaired people have to touch the obstacle with the guide cane, before they get the position of the obstacle and avoid it. Sometimes touching the obstacle, the danger already has occurred. The two methods mentioned above have plenty of inconveniences. Using computer vision can avoid a lot of inconveniences, so how to detect the obstacles efficiently is an important issue. In recent years, there are a lot of developments and advancements in computer vision. Many scholars have proposed a lot of obstacle detection methods. Obstacle detection can be classified into two categories. One is based non-depth information, and the other is based on the depth information.

For the first category, there are many proposed methods like [1-4]. An object detection algorithm based on edge and motion has been proposed [1]. This method uses the information of motion to find out the dynamic obstacles. And it uses the information of edge to search static obstacles. Finally, they combine previous information with free space detection to find out the position of the obstacles. An obstacle detection algorithm by a single camera is proposed [2]. This work uses edge detection to segment objects. But this method needs without complex texture on the surface of the ground. In [3], an obstacle detection based on saliency map has been proposed. This work uses threshold value to obtain the position of the obstacles. The execution of the proposed method requires a few obstacles in the environment. An obstacle detection based on grayscale image has been proposed [4]. It searches the region of interest in grayscale image, and then finds out the location of obstacles. But the proposed method works on the grayscale image, so it is affected easily by illumination.

The methods of obstacles detection in the second category have [5-7]. An obstacle detection algorithm based on U-V disparity map analysis is presented in [5]. This work combines straight line fitting and the standard Hough transform to find out the location of the obstacles. But the U-V disparity map is generated by two webcams. The illumination affects the performance of the system. In [6], the depth image is obtained by 3D camera. This work combined U-V disparity map to find out the location of the obstacles. An obstacles detection based on Kinect sensor has been proposed in [7]. This work uses Kinect sensor to obtain color images and depth images. It respectively uses edge detection for color images and depth images, and then processes these edge images by morphology. Finally, it fuses the results of the two images to get the position of the obstacles. But the color image of this work is still affected by illumination. So the experimental environment is restricted

The paper is organized as followings: Section Ⅱ is the system overview and the internal details of the system. Section III is the experimental results in the different environment. Section IV is conclusions.

2. Our Method

2.1. System Architecture

Process of our system is as follows: First, to capture the depth image. Second, remove the noise of depth image by using morphological. Third, extract the edge of the different objects and remove it. Fourth, remove the information of the ground. Fifth, label tags on the different objects. Finally, combine with the color image to get the positioning of the obstacle and use depth information to obtain the distance between the obstacles and the Kinect sensor. The system flowchart is Fig. 1.

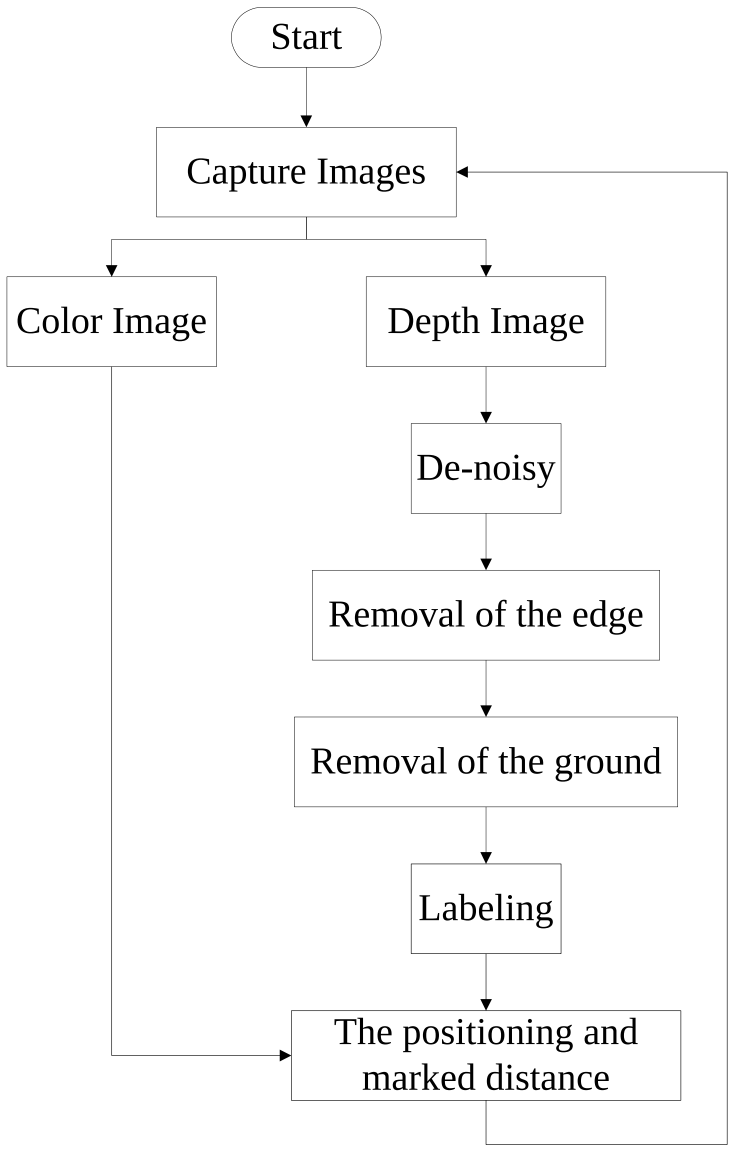
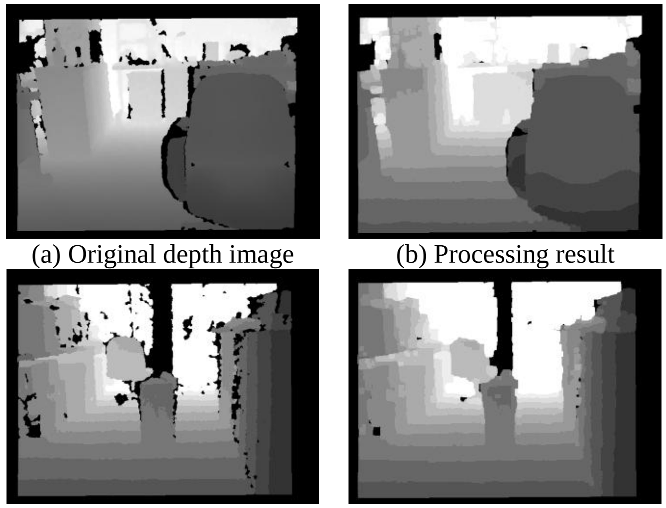


Figure 1. The system flowchart.



(a) Original depth image (b) Processing result
(c) Original depth image (d) Processing result

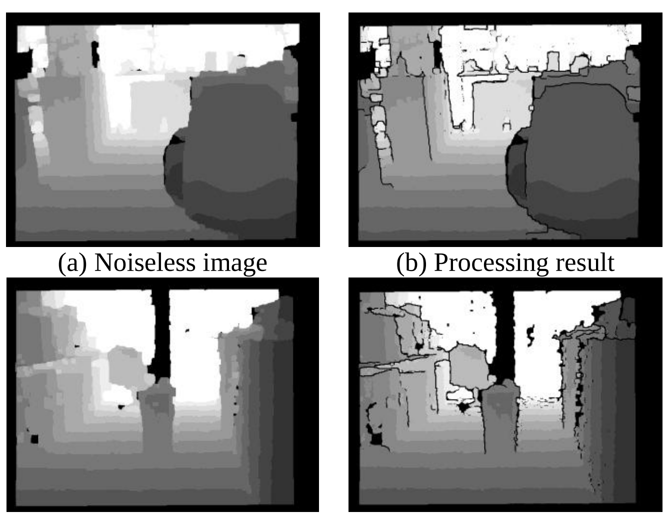
Figure 2. Noise Removal.

2.2. Noise Removal

Because of Kinect hardware limitations, the depth image will have broken phenomenon. In order to make the depth image more complete, we apply to some simple morphology processing. In this paper, we use erosion and expansion to repair the black broken area. By the following Fig. 2, we can see that the processed depth images are better than non-processed depth images.

2.3. Removal of the edge

In the depth image, the depth represents the distance of the objects and sensor. By depth variation, we can know whether these obstacles are the same object. Depth variation of the same object usually is not too intense. In the different objects, the relationship of the distance will cause depth value varies strongly. In this paper, in order to strengthen the characteristics of different objects, we remove the strong edge. And there are so many edge detection methods, as Roberts, Prewitt, Sobel, Laplace and Canny. In this paper, we detect edge by the following function (1). The process result is shown in Fig. 3.



(a) Noiseless image (b) Processing result
(c) Noiseless image (d) Processing result

Figure 3. Removal of the edge.

$$P(x) = \begin{cases} 0, & \text{if } \sum_{x_1 \in x_n} |P(x) - P(x_1)| \geq Th \\ \text{unchange}, & \text{others} \end{cases} \quad (1)$$

Here P(i,j) represents the pixel value of the coordinates (i, j). Th represents the threshold.

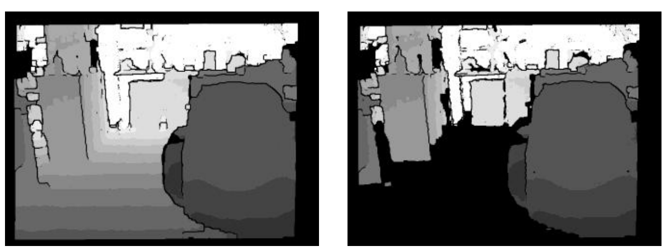
2.4. Removal of the ground

If we directly use the connected component labeling or other labeling methods to label tags, the ground and obstacles will not be separated because the junction of the ground and obstacles have the same depth value. In this time, we need to remove the information of the ground firstly. The RANSAC plane fitting [8] is used to find ground plane in the 3D space. Because the sensor cannot be fixed, the calculation of the ground information needs to repeat iterations. In order to improve system speed, we use the following information filtered floor. (1)Ground is usually relatively flat. (2)And we observe from the information of depth, the gray value will be from large to small (from far to near). Using this feature, we can extract the suspicious planes which meet the conditions. The suspicious plane information needs further processing. We just need the large areas of the ground. So we apply this method proposed in [9]. By searching for the inner and outer contours, we can find out regions and sizes of the different suspicious planes. And then remove the plane which has a large area. The processing result is shown in Fig. 4.

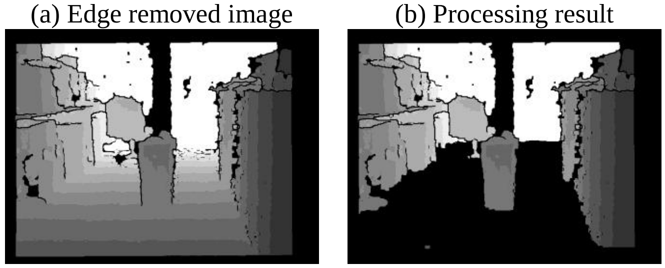
2.5. Labeling

The sensing range of Kinect is 0.8 to 4.0 meters. When the range farther than the max distance, it will not be able to distinguish the distance. So, we need to remove the far information. In order to measure distances accurately, we retain the information within 3 meters.

And then, we label different tags on different objects. The general labeling methods have 8 connected component labeling and region growing. But the tag harmonization of connected component labeling needs a lot of iterations, because of the complex shape of the connected area. In order to make the system faster, we use region growing [10]. The initialization of traditional region growing has to sprinkle some seeds in the image. If the distribution of the sprinkled seeds is not good, it will cause the growth result be unsatisfactory. So the choice of the initial position of the seeds has made some improvements in the propose system. We combine with the information of the object edges. Because the previous steps remove the edge information, each object is surrounded by black color. We use the following formula (2) to select the coordinates of initial seeds and utilize these coordinates to execute region growing. It is able to ensure that each object has an initial seed, and the growth of the place

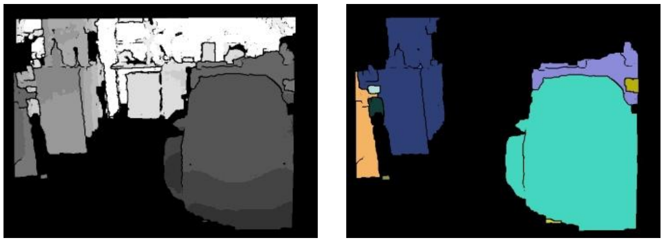


(a) Edge removed image (b) Processing result

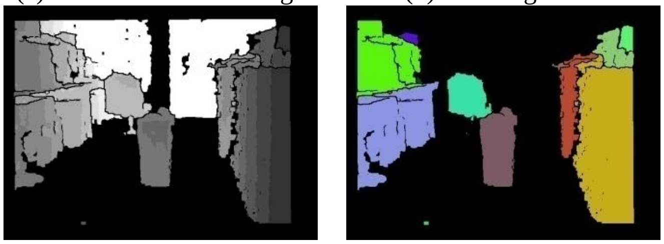


(c) Edge removed image (d) Processing result

Figure 4. Removal of the ground.



(a) Ground removed image (b) Labeling result



(c) Ground removed image (d) Labeling result

Figure 5. Labeling.

would not be repeated treatment. Therefore, we propose system reduce the amount of computation. The processing result is shown in Fig. 5.

$$S(i,j) = \begin{cases} (i,j), \text{if } [\,(P(i-1,j-1)=0) \\ \quad \wedge (P(\;i\;,j-1)=0 \\ \quad \wedge (P(i+1,j-1)=0 \\ \quad \wedge (P(i-1,\;j\;)=0 \\ \quad \wedge (P(\;i\;,\;j\;)\neq 0)\,] \\ \text{not seed}, \quad \text{others} \end{cases} \qquad (2)$$

Here S(i,j) represents the seed coordinates. P(i,j) represents the pixel value at the coordinates (i, j).

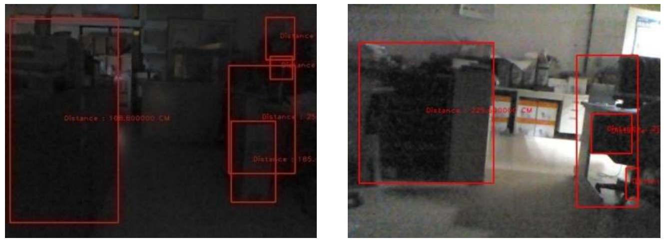
3. Experimental result

The experimental image capture tool is Microsoft Kinect sensor. Experimental platform is Windows 7. Programming language is Visual C++. The image resolution is 640×480. The depth image capture rate is 30 frames per second. The sensing range is 0.8 to 3.0 meters.

The experiments are tested in the different brightness of the indoor environment. Fig. 6(a) and 6(b) are the results in the bright indoor environment. Fig. 6(c) and 6(d) are the results in the low-light indoor environment.

When obstacles appear in front of the user, the system will give the user distance information of the obstacle by voice.

(a) Bright indoor (b) Bright indoor



(c) Low-light indoor (d) Low-light indoor

Figure 6. Obstacle detection results.

4. References

[1] Ma G., et al., "A Real-Time Rear View Camera Based Obstacle Detection," 12th IEEE International Conference on Intelligent Transportation Systems, pp 1-6, September 2009.Author1_Name, Author2_Name, "Exploring the bounds of Web latency reduction from caching and prefetching", In Proceeding(s) of the AICIT Symposium(Conference Title) on Internet Technologies, pp.13-22, 2009.

[2] Yankun, Z., Hong, C., Weyrich, N., "A single camera based rear obstacle detection system," Intelligent Vehicles Symposium, 2011 IEEE , vol., no.,pp.485-490, 5-9 June 2011.

[3] Chen, L., Guo, B.L., Sun, W., "Obstacle Detection System for Visually Impaired People Based on Stereo Vision," Genetic and Evolutionary Computing, 2010 Fourth International Conference on, vol., no., pp.723-726, 13-15 Dec. 2010.

[4] Ying Jie and Song Yanbin "Obstacle Detection of a Novel Travel Aid for Visual Impaired People," Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on, vol., no., pp. 362-364, 26-27 Aug. 2012.

[5] Zhang M., et al., "An Obstacle Detection Algorithm Based on U-V Disparity Map Analysis," Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on, vol., no., pp. 763-766, 17-19 Dec. 2010.

[6] Yuan Gao, et al., "Obstacle Detection with 3D Camera Using U-V Disparity," Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on, vol., no., pp.239- 242, 9-11 May 2011.

[7] Jinwook Choi, et al., "Rear Obstacle Detection System Based on Depth From Kinect," Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, vol., no., pp.98-101, 16-19 Sept. 2012.

[8] C. Zheng and R. Green, "Feature Recognition and Obstacle Detection for Drive Assistance in Indoor Environments," University of Canterbury, Christchurch, New Zealand.

[9] Suzuki S, Abe K., "Topological structural analysis of digitized binary images by border following," (J). Computer Vision, Graphics, and Image Processing, 1985, 30(1):32-46.

[10] R. Adams, and L. Bischof, "Seeded region growing," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 16, no. 6, pp. 641-647, June, 1994.